

Exercice 1

Ecrire un script "boucle.sh" qui prend comme arguments 3 valeurs entières: la première est la valeur de départ (X), la seconde est la valeur finale (Y) et la 3ième est l'incrément (I). Il affiche les valeurs suivantes

X

X+I

X+2I

...

Y

Si l'utilisateur rentre deux arguments, l'incrément par défaut sera 1. En présence d'un seul argument, la valeur de départ X est par défaut 1. En absence de tous paramètres, le script affiche tous les chiffres entre 1 et 30 avec un pas égal à 1.

Exercice 2

Ecrire un script qui demande à l'utilisateur de rentrer un nombre entier n, et qui affiche la séquence de x-5 à x+5.

Exercice 3

Ecrire un script qui prend plusieurs arguments et qui l'affiche un par ligne.

Exercice 4

Ecrire un script "test-fichier.sh", qui précisera le type du fichier passé en argument, ses permissions d'accès pour l'utilisateur. Si le script ne reçoit aucun paramètre, il devra afficher un message d'erreur avec la fonction Usage.

Exemple de résultats :

Le fichier /etc est un répertoire

"/etc" est accessible par root en lecture écriture exécution

Le fichier /etc/smb.conf est un fichier ordinaire qui n'est pas vide

"/etc/smb.conf" est accessible par toto en lecture.

Exercice 5

Ecrire un script "taille.sh" qui renvoie la taille d'un fichier en octets.

```
$ ls -l filename
```

```
-rw-r--r-- 1 etu etu 636 Nov 05 19:30 filename
```

```
$ ./taille.sh filename
```

```
636
```

Indication :

Utiliser la commande **ls -l, tr** et la commande **cut** pour extraire un champ d'une ligne.

La commande `tr` accepte 3 options

- c : complément d'une liste
- d : delete/supprimer
- s : remplacement de caractères successifs par un seul.

Exemples :

```
tr -s '\n' < inputtextfile > outputfile
tr -s ' ' < inputtextfile
echo "a test" | tr t p
echo "a test" | tr aest 1234
echo "a test" | tr -d t
tr A-Z a-z < file1
```

La commande "**cut**" permet d'extraire une partie des lignes d'un fichier. Elle prend 3 options:

- c2,5 : extraire les caractères 2, 3, 4,5 de chaque ligne
- f1,5 : extraire les champs de 1 à 5 de chaque ligne
- d : spécifie le délimiteur

Exemples:

```
$ echo "hello" | cut -c 2-4
ell
$ echo "he ll o wo" | cut -f2,3 d' '
ll o
cut -f1 -d': ' < /etc/passwd
```

Exercice 5

Ecrire un script dans lequel deux nombres opérands et un signe opérateur (+-*/) devront être donnés en paramètres, ou saisis en cas d'absence. Le script doit réaliser l'opération souhaitée.

```
$/cal 11 * 8
```

Exercice 7

Ecrire un script qui boucle en demandant à l'utilisateur de saisir deux variables, affiche leur somme, leur différence, leur produit, leur quotient et leur exponentiation. Le programme s'arrête lorsque l'utilisateur saisit un point.

Exercice 8

Ecrire un script qui prend un nombre indéterminé des arguments et qui calcule leur somme et la valeur moyenne.

Exercice 9

Ecrire un script qui prend un argument et en affiche la factorielle. Vérifier que le nombre de paramètres est correct.

Exercice 10

Écrire un script qui affiche la liste de login des utilisateurs dans `/etc/passwd` ayant un UID supérieur ou égal à 1000.

Modifier le script précédent pour :

1. Demander un login
2. Vérifier l'existence de l'utilisateur
3. Afficher l'UID, GID et le répertoire du travail d'un utilisateur existant
4. Quitter

Exercice 11

Écrire une fonction qui retourne le nombre de jours dans chaque mois de l'année.

Modifier le script pour demander à l'utilisateur un nombre entre 1 et 12 et qui appelle la fonction précédente pour afficher le nombre de jours dans ce mois.

Exercice 12

Écrire un script qui prend en paramètre (ou en saisie en cas d'absence de paramètres) une valeur décimale et qui doit la convertir en binaire, octal et hexadécimal.

Exercice 13

Écrire un script qui permet de renommer tous les fichiers dans le répertoire actuel (sauf le script) en ajoutant l'extension `.bak` au nom de chaque fichier.

Exercice 14

Écrire un script qui prend un argument un nombre entier (n) inférieur ou égal à 12 et qui fait appel à une fonction pour calculer 2^n . Ensuite, il génère 10 nombres aléatoires entre 1 et 2^n et affiche (10 nombres et la valeur de 2^n)